Python for Data processing

Lecture 5:
Plotting and time series

Gleb Ivashkevich

What we already know

- **Numpy**
- PyTorch
- **pandas**: series and dataframes, reading files, **`groupby`**, joins

Today

- plotting with Matplotlib and Seaborn
- time series in **pandas** (**DateTimeIndex** and others)

matplotlib: plotting with Python

matplotlib

Plotting library for Python:

- relies on **numpy** arrays (we'll see, how this works in **pandas**)
- a lot of plotting options, output formats and UI toolkits
- publication ready images
- low level

matplotlib figures

It all starts with **Figure** (explicitly or implicitly)

Figure:

- can have size
- multiple subplots
- other properties

→let's try it out!

matplotlib plots: line

plt.plot to plot simple y(x) for two (or single!) arrays:

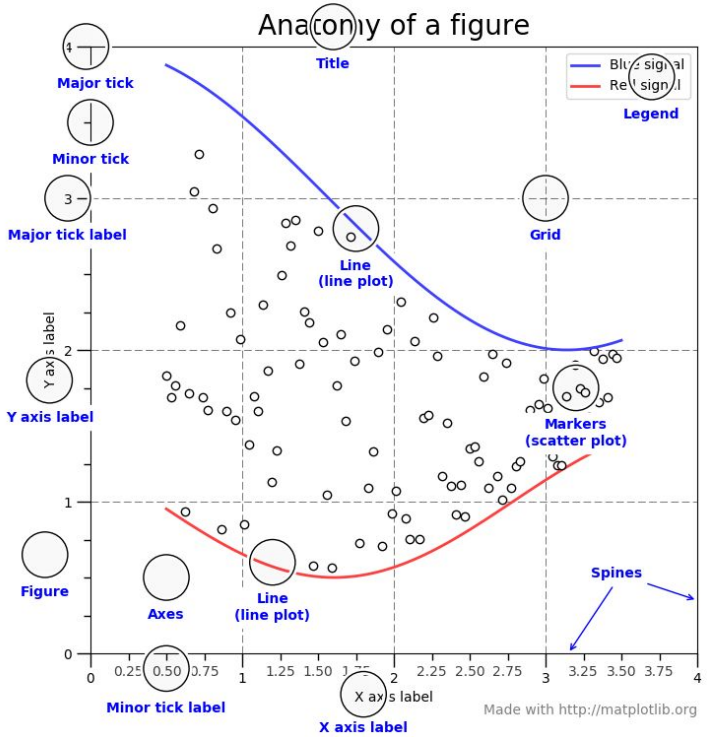
- you can change line appearance
- plot multiple lines on a single figure (axes)

Hint:

- use predefined styling

→let's try it out!

matplotlib figures: elements



matplotlib plots: scatter

plt.scatter to plot simple (x, y) pairs:

- you can change markers appearance
- point-wise color and size

→let's try it out!

matplotlib plots: histogram

plt.hist to plot distribution of x:

- select bin size and number of bins
- stacked histograms

→let's try it out!

matplotlib figures: box plots

plt.boxplot to to get another view of variable(-s) distribution

→let's try it out!

matplotlib figures: subplots

Each figure can contain multiple plots:

- use **plt.subplot(rows, columns, plot_number)**
- or **ax = fig.add_subplot(rows, columns, plot_number)**

→let's try it out!

Seaborn

Stylish plotting:

- based on matplotlib
- many additional types of plots
- styling

→let’s try it out!

Time series

Time series: definition

→ time-ordered sequence of values (multivariate): $s_{\alpha}(t_k)$

→ $t_0 < t_1 < t_2 \ldots < t_N$

→ may be unevenly spaced,

→ very large Δt may be treated as a gap.

Time series: examples

→ EEG, ECG and other physiological signals (~1000-100 Hz),

→ motion sensors (~100-10 Hz),

→ factory sensors measurements (~10-10^{-1} Hz),

→ number of customers in a store per hour (10^{-1}-10^{-2} Hz).

Event stream: definition

→ a set of entities, having some attributes and indexed by some form of timestamp: $\mathbf{E}(t; \boldsymbol{a}_0, \boldsymbol{a}_1, \ldots)$

→ Individual events may be not related to each other,

→ aggregates from event stream may be represented as time series.

Event stream: examples

→ **equipment failures on a factory:**

equipment operational parameters, equipment condition (age, last maintenance, etc.), type of product, etc.

→ **car accidents:**

location, speed, road type, weather, etc.

→ **click stream:**

ad details, user agent data, user location, etc.

Event stream exploration

→ **problem:**

get insights from raw events

→ **approach:**

statistics, restructure to time series, plot different aggregates

→ **dataset:**

1.6 million UK traffic accidents

https://www.kaggle.com/daveianhickey/2000-16-traffic-flow-england-scotland-wales/data

Dates in pandas

→ pandas is efficient and powerful in **handling datetimes**

→ **shift** operations

→ **rolling** operations

→ **resampling** operations

→ datetime based **joins**

Dates in pandas

- each datetime column has **accessor called `.dt`** to efficiently query datetime components and calculations
- **parsing** from strings and **time zones** are handled (almost) transparently

What we already know

- **Jupyter**
- numpy
- PyTorch
- pandas
- Matplotlib (+ some Seaborn)

Next

- exploratory data analysis
- tools, tips and tricks
- overview of data science and ML landscape

questions?